IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the Application of: GARY L. GRAUNKE, ET AL. Application No.: Filed: For: **An Apparatus and Method for Memory Encryption with Reduced Decryption Latency**	Art Group: Examiner:

INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. §1.97

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the duty of disclosure, enclosed is a copy of Information Disclosure Statement by Applicant (form PTO/SB/08), which is being submitted concurrently with the Utility Application. It is respectfully requested that the cited references be considered and that the enclosed copy of PTO/SB/08 be initialed by the Examiner to indicate such consideration and a copy thereof returned to applicant(s). Copies of the references cited on PTO/SB/08 are enclosed herewith.

The submission of this Information Disclosure Statement is not to be construed as a representation that a search has been made in the subject application and is not to be construed as an admission that the information cited in this statement is material to patentability.

Please charge any fees due to Deposit Account 02-2666. A duplicate copy of the Fee Transmittal (PTO/SB/17) is enclosed for this purpose.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: June 25, 2003

Joseph Lutz, Reg. No. 43,765

12400 Wilshire Blvd., 7th Floor
Los Angeles, California 90025
(310) 207-3800

Substitute for form 1449A/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				Application Number	
(use as many sheets as necessary)				Filing Date	
				First Named Inventor	Gary L. Graunke
				Art Unit	
				Examiner Name	
Sheet	1	**of**	1	Attorney Docket Number	42P16433

OTHER ART - NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
		Dworkin, Morris "Recommendation for Block Cipher Modes of Operation," *Methods and Techniques*, COMPUTER SECURITY, NIST, 2001 Edition, pp. i-vii, 9-16	

Examiner Signature		Date Considered	

*Examiner: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication.

[1]Applicant's unique citation designation number. [2]Applicant is to place a check mark here if English language Translation is attached.

Based on PTO/SB/08B (05-03) as modified by Blakely, Solokoff, Taylor & Zafman (wlr) 05/02/2003.
Send To: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

NIST Special Publication 800-38A
2001 Edition

Recommendation for Block Cipher Modes of Operation

Methods and Techniques

Morris Dworkin

NIST
National Institute of Standards and Technology
Technology Administration
U.S. Department of Commerce

COMPUTER SECURITY

COMPUTER SECURITY

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

December 2001

U.S. Department of Commerce
Donald L. Evans, Secretary

Technology Administration
Phillip J. Bond, Under Secretary of Commerce for Technology

National Institute of Standards and Technology
Arden L. Bement, Jr., Director

Reports on Information Security Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security, and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-38A 2001 ED
Natl. Inst. Stand. Technol. Spec. Publ. 800-38A 2001 ED, 66 pages (December 2001)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 2001

For sale by the Superintendent of Documents, U.S. Government Printing Office
Internet: bookstore.gpo.gov — Phone: (202) 512-1800 — Fax: (202) 512-2250
Mail: Stop SSOP, Washington, DC 20402-0001

Abstract

This recommendation defines five confidentiality modes of operation for use with an underlying symmetric key block cipher algorithm: Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). Used with an underlying block cipher algorithm that is approved in a Federal Information Processing Standard (FIPS), these modes can provide cryptographic protection for sensitive, but unclassified, computer data.

KEY WORDS: Computer security; cryptography; data security; block cipher; encryption; Federal Information Processing Standard; mode of operation.

Table of Contents

Table of Figures

6 Block Cipher Modes of Operation

The mathematical specifications of the five modes are given in Sections 6.1-6.5, along with descriptions, illustrations, and comments on the potential for parallel processing.

6.1 The Electronic Codebook Mode

The Electronic Codebook (ECB) mode is a confidentiality mode that features, for a given key, the assignment of a fixed ciphertext block to each plaintext block, analogous to the assignment of code words in a codebook. The Electronic Codebook (ECB) mode is defined as follows:

ECB Encryption: $C_j = CIPH_K(P_j)$ for $j = 1 \ldots n$.

ECB Decryption: $P_j = CIPH^{-1}_K(C_j)$ for $j = 1 \ldots n$.

In ECB encryption, the forward cipher function is applied directly and independently to each block of the plaintext. The resulting sequence of output blocks is the ciphertext.

In ECB decryption, the inverse cipher function is applied directly and independently to each block of the ciphertext. The resulting sequence of output blocks is the plaintext.

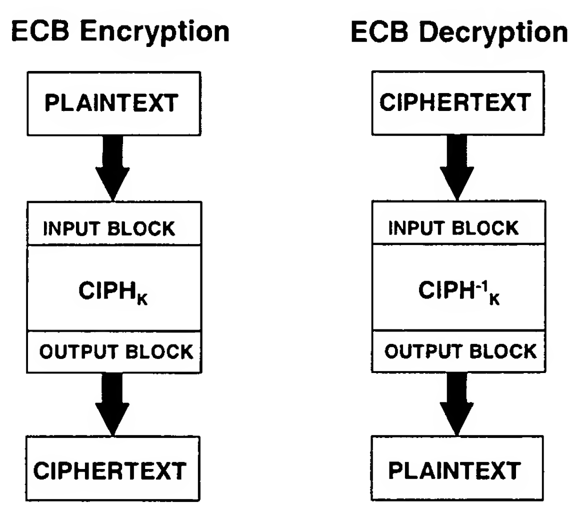

Figure 1: The ECB Mode

In ECB encryption and ECB decryption, multiple forward cipher functions and inverse cipher functions can be computed in parallel.

In the ECB mode, under a given key, any given plaintext block always gets encrypted to the

same ciphertext block. If this property is undesirable in a particular application, the ECB mode should not be used.

The ECB mode is illustrated in Figure 1.

6.2 The Cipher Block Chaining Mode

The Cipher Block Chaining (CBC) mode is a confidentiality mode whose encryption process features the combining ("chaining") of the plaintext blocks with the previous ciphertext blocks. The CBC mode requires an IV to combine with the first plaintext block. The IV need not be secret, but it must be unpredictable; the generation of such IVs is discussed in Appendix C. Also, the integrity of the IV should be protected, as discussed in Appendix D. The CBC mode is defined as follows:

CBC Encryption: $C_1 = CIPH_K(P_1 \oplus IV)$;
$C_j = CIPH_K(P_j \oplus C_{j-1})$ for $j = 2 \ldots n$.

CBC Decryption: $P_1 = CIPH^{-1}_K(C_1) \oplus IV$;
$P_j = CIPH^{-1}_K(C_j) \oplus C_{j-1}$ for $j = 2 \ldots n$.

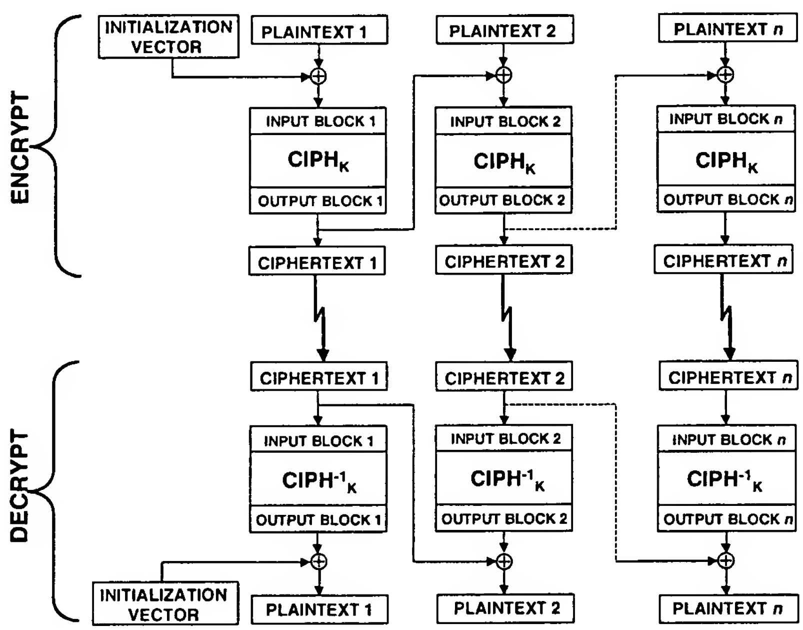

Figure 2: The CBC Mode

In CBC encryption, the first input block is formed by exclusive-ORing the first block of the plaintext with the IV. The forward cipher function is applied to the first input block, and the

resulting output block is the first block of the ciphertext. This output block is also exclusive-ORed with the second plaintext data block to produce the second input block, and the forward cipher function is applied to produce the second output block. This output block, which is the second ciphertext block, is exclusive-ORed with the next plaintext block to form the next input block. Each successive plaintext block is exclusive-ORed with the previous output/ciphertext block to produce the new input block. The forward cipher function is applied to each input block to produce the ciphertext block.

In CBC decryption, the inverse cipher function is applied to the first ciphertext block, and the resulting output block is exclusive-ORed with the initialization vector to recover the first plaintext block. The inverse cipher function is also applied to the second ciphertext block, and the resulting output block is exclusive-ORed with the first ciphertext block to recover the second plaintext block. In general, to recover any plaintext block (except the first), the inverse cipher function is applied to the corresponding ciphertext block, and the resulting block is exclusive-ORed with the previous ciphertext block.

In CBC encryption, the input block to each forward cipher operation (except the first) depends on the result of the previous forward cipher operation, so the forward cipher operations cannot be performed in parallel. In CBC decryption, however, the input blocks for the inverse cipher function, i.e., the ciphertext blocks, are immediately available, so that multiple inverse cipher operations can be performed in parallel.

The CBC mode is illustrated in Figure 2.

6.3 The Cipher Feedback Mode

The Cipher Feedback (CFB) mode is a confidentiality mode that features the feedback of successive ciphertext segments into the input blocks of the forward cipher to generate output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The CFB mode requires an IV as the initial input block. The IV need not be secret, but it must be unpredictable; the generation of such IVs is discussed in Appendix C.

The CFB mode also requires an integer parameter, denoted s, such that $1 \leq s \leq b$. In the specification of the CFB mode below, each plaintext segment ($P^{\#}_j$) and ciphertext segment ($C^{\#}_j$) consists of s bits. The value of s is sometimes incorporated into the name of the mode, e.g., the 1-bit CFB mode, the 8-bit CFB mode, the 64-bit CFB mode, or the 128-bit CFB mode.

The CFB mode is defined as follows:

CFB Encryption:	$I_1 = IV;$	
	$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1}$	for $j = 2 \ldots n;$
	$O_j = CIPH_K(I_j)$	for $j = 1, 2 \ldots n;$
	$C^{\#}_j = P^{\#}_j \oplus MSB_s(O_j)$	for $j = 1, 2 \ldots n.$
CFB Decryption:	$I_1 = IV;$	
	$I_j = LSB_{b-s}(I_{j-1}) \mid C^{\#}_{j-1}$	for $j = 2 \ldots n;$

$$O_j = CIPH_K(I_j) \qquad \text{for } j = 1, 2 \ldots n;$$
$$P^\#_j = C^\#_j \oplus MSB_s(O_j) \qquad \text{for } j = 1, 2 \ldots n.$$

In CFB encryption, the first input block is the IV, and the forward cipher operation is applied to the IV to produce the first output block. The first ciphertext segment is produced by exclusive-ORing the first plaintext segment with the s most significant bits of the first output block. (The remaining b-s bits of the first output block are discarded.) The b-s least significant bits of the IV are then concatenated with the s bits of the first ciphertext segment to form the second input block. An alternative description of the formation of the second input block is that the bits of the first input block circularly shift s positions to the left, and then the ciphertext segment replaces the s least significant bits of the result.

The process is repeated with the successive input blocks until a ciphertext segment is produced from every plaintext segment. In general, each successive input block is enciphered to produce an output block. The s most significant bits of each output block are exclusive-ORed with the corresponding plaintext segment to form a ciphertext segment. Each ciphertext segment (except the last one) is "fed back" into the previous input block, as described above, to form a new input block. The feedback can be described in terms of the individual bits in the strings as follows: if $i_1i_2...i_b$ is the jth input block, and $c_1c_2...c_s$ is the jth ciphertext segment, then the $(j+1)^{\text{th}}$ input block is $i_{s+1}i_{s+2}...i_b\ c_1c_2...c_s$.

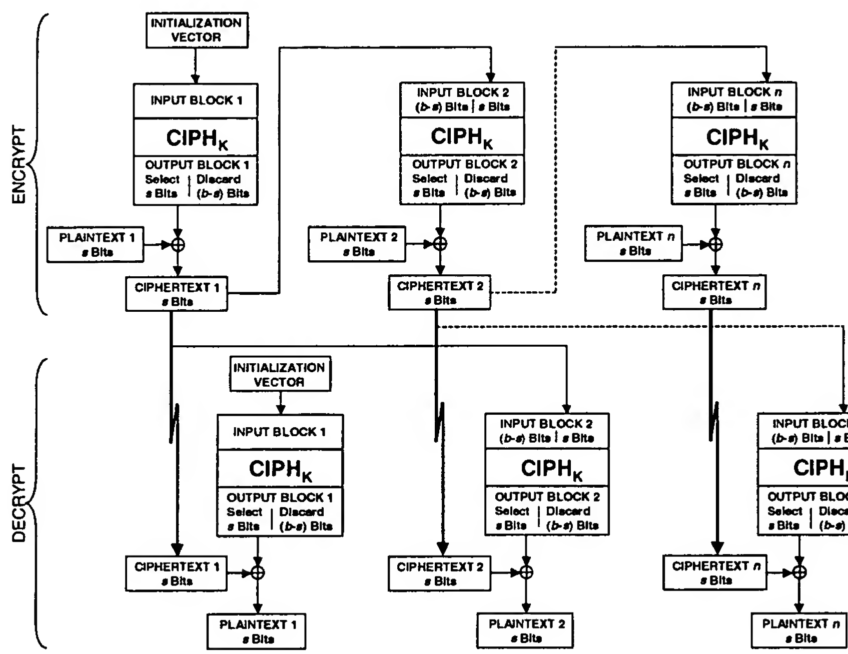

Figure 3: The CFB Mode

In CFB decryption, the IV is the first input block, and each successive input block is formed as in CFB encryption, by concatenating the b-s least significant bits of the previous input block with

the s most significant bits of the previous ciphertext. The *forward cipher* function is applied to each input block to produce the output blocks. The s most significant bits of the output blocks are exclusive-ORed with the corresponding ciphertext segments to recover the plaintext segments.

In CFB encryption, like CBC encryption, the input block to each forward cipher function (except the first) depends on the result of the previous forward cipher function; therefore, multiple forward cipher operations cannot be performed in parallel. In CFB decryption, the required forward cipher operations can be performed in parallel if the input blocks are first constructed (in series) from the IV and the ciphertext.

The CFB mode is illustrated in Figure 3.

6.4 The Output Feedback Mode

The Output Feedback (OFB) mode is a confidentiality mode that features the iteration of the forward cipher on an IV to generate a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The OFB mode requires that the IV is a nonce, i.e., the IV must be unique for each execution of the mode under the given key; the generation of such IVs is discussed in Appendix C. The OFB mode is defined as follows:

OFB Encryption:	$I_1 = IV;$	
	$I_j = O_{j-1}$	for $j = 2 \ldots n;$
	$O_j = CIPH_K(I_j)$	for $j = 1, 2 \ldots n;$
	$C_j = P_j \oplus O_j$	for $j = 1, 2 \ldots n\text{-}1;$
	$C^*_n = P^*_n \oplus MSB_u(O_n).$	
OFB Decryption:	$I_1 = IV;$	
	$I_j = O_{j-1}$	for $j = 2 \ldots n;$
	$O_j = CIPH_K(I_j)$	for $j = 1, 2 \ldots n;$
	$P_j = C_j \oplus O_j$	for $j = 1, 2 \ldots n\text{-}1;$
	$P^*_n = C^*_n \oplus MSB_u(O_n).$	

In OFB encryption, the IV is transformed by the forward cipher function to produce the first output block. The first output block is exclusive-ORed with the first plaintext block to produce the first ciphertext block. The forward cipher function is then invoked on the first output block to produce the second output block. The second output block is exclusive-ORed with the second plaintext block to produce the second ciphertext block, and the forward cipher function is invoked on the second output block to produce the third output block. Thus, the successive output blocks are produced from applying the forward cipher function to the previous output blocks, and the output blocks are exclusive-ORed with the corresponding plaintext blocks to produce the ciphertext blocks. For the last block, which may be a partial block of u bits, the most significant u bits of the last output block are used for the exclusive-OR operation; the remaining b-u bits of the last output block are discarded.

In OFB decryption, the IV is transformed by the *forward cipher* function to produce the first

output block. The first output block is exclusive-ORed with the first ciphertext block to recover the first plaintext block. The first output block is then transformed by the forward cipher function to produce the second output block. The second output block is exclusive-ORed with the second ciphertext block to produce the second plaintext block, and the second output block is also transformed by the forward cipher function to produce the third output block. Thus, the successive output blocks are produced from applying the forward cipher function to the previous output blocks, and the output blocks are exclusive-ORed with the corresponding ciphertext blocks to recover the plaintext blocks. For the last block, which may be a partial block of u bits, the most significant u bits of the last output block are used for the exclusive-OR operation; the remaining b-u bits of the last output block are discarded.

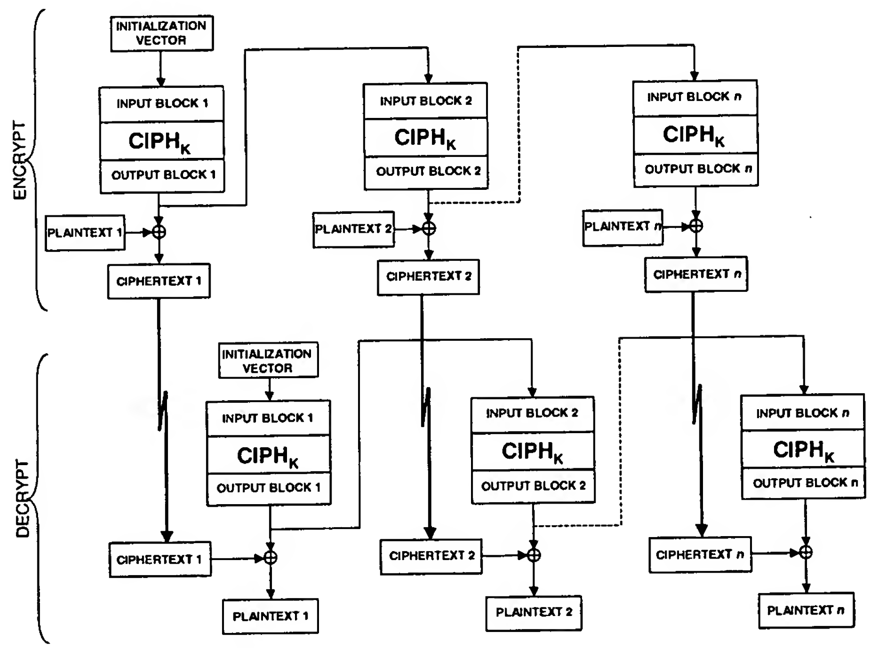

Figure 4: The OFB Mode

In both OFB encryption and OFB decryption, each forward cipher function (except the first) depends on the results of the previous forward cipher function; therefore, multiple forward cipher functions cannot be performed in parallel. However, if the IV is known, the output blocks can be generated prior to the availability of the plaintext or ciphertext data.

The OFB mode requires a unique IV for every message that is ever encrypted under the given key. If, contrary to this requirement, the same IV is used for the encryption of more than one message, then the confidentiality of those messages may be compromised. In particular, if a plaintext block of any of these messages is known, say, the jth plaintext block, then the jth output of the forward cipher function can be determined easily from the jth ciphertext block of the message. This information allows the jth plaintext block of any other message that is encrypted

using the same IV to be easily recovered from the jth ciphertext block of that message.

Confidentiality may similarly be compromised if *any* of the input blocks to the forward cipher function for the encryption of a message is designated as the IV for the encryption of another message under the given key.

The OFB mode is illustrated in Figure 4.

6.5 The Counter Mode

The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The sequence of counters must have the property that each block in the sequence is different from every other block. This condition is not restricted to a single message: across all of the messages that are encrypted under the given key, all of the counters must be distinct. In this recommendation, the counters for a given message are denoted $T_1, T_2, \ldots, T_n$. Methods for generating counters are discussed in Appendix B. Given a sequence of counters, $T_1, T_2, \ldots, T_n$, the CTR mode is defined as follows:

CTR Encryption:

$$O_j = CIPH_K(T_j) \quad \text{for } j = 1, 2 \ldots n;$$
$$C_j = P_j \oplus O_j \quad \text{for } j = 1, 2 \ldots n\text{-}1;$$
$$C^*_n = P^*_n \oplus MSB_u(O_n).$$

CTR Decryption:

$$O_j = CIPH_K(T_j) \quad \text{for } j = 1, 2 \ldots n;$$
$$P_j = C_j \oplus O_j \quad \text{for } j = 1, 2 \ldots n\text{-}1;$$
$$P^*_n = C^*_n \oplus MSB_u(O_n).$$

In CTR encryption, the forward cipher function is invoked on each counter block, and the resulting output blocks are exclusive-ORed with the corresponding plaintext blocks to produce the ciphertext blocks. For the last block, which may be a partial block of u bits, the most significant u bits of the last output block are used for the exclusive-OR operation; the remaining b-u bits of the last output block are discarded.

In CTR decryption, the forward cipher function is invoked on each counter block, and the resulting output blocks are exclusive-ORed with the corresponding ciphertext blocks to recover the plaintext blocks. For the last block, which may be a partial block of u bits, the most significant u bits of the last output block are used for the exclusive-OR operation; the remaining b-u bits of the last output block are discarded.

In both CTR encryption and CTR decryption, the forward cipher functions can be performed in parallel; similarly, the plaintext block that corresponds to any particular ciphertext block can be recovered independently from the other plaintext blocks if the corresponding counter block can be determined. Moreover, the forward cipher functions can be applied to the counters prior to the availability of the plaintext or ciphertext data.

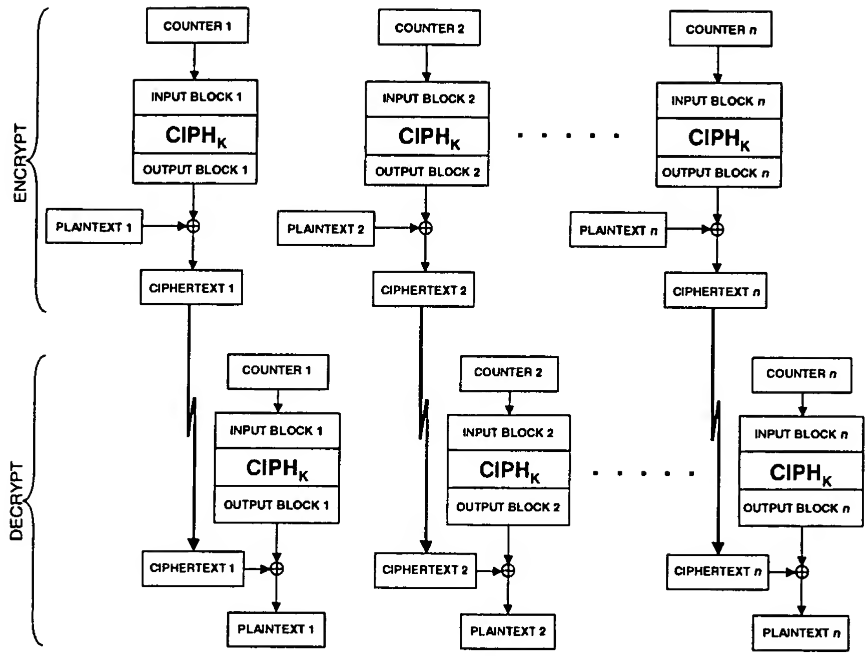

Figure 5: The CTR Mode

The CTR mode is illustrated in Figure 5.